Projet

Problème des sous-séquences commune les plus longue

~ Longest Common Subsequence problem (LCS)

Le problème LCS a des applications dans de nombreux domaines, la bioinformatique en fait partie. Il peut être résolu par programmation dynamique, mais cette approche devient peu pratique pour les grandes tailles de n. D'autres versions restreintes ont été abordées dans la littérature, étant les plus importantes.

Définition: Étant donné un ensemble S = {s_1 , s_2 , . . . , s_n } de n chaînes sur un alphabet fini, le problème LCS consiste à trouver la chaîne "t" la plus longue qui soit une sous-séquence (dans le même ordre) de toutes les chaînes de S. On considère que la chaîne "t" est obtenue en supprimant des caractères d'une chaîne s.

Exemple 1: Soit S = {S_1 , S_2 } un ensemble de trois chaînes sur l'alphabet fini α = {A,G,I,K, M,Y,V} tels que chaque lettre représente un acide aminé; *Alanine* (A); *Glycine* (G); *Isoleucine* (I); *Lysine* (K); *Méthionine* (M); *Tyrosine* (Y); *Valine* (V); pouvant constituer différentes protéines.

S_1: MAYAGIVK ; S_2: MAKAGVI

⇒ La longueur du LCS est de **5**
La 5-LCS est : MAAGI

Exemple 2: Soit S = {S_1 , S_2 , S_3 } un ensemble de trois chaînes sur l'alphabet fini α = {A,C,G,T} tels que *Adénine* (A), *Thymine* (T), *Cytosine* (C) et *Guanine* (G) sont les nucléotides constituant l'ADN.

S_1: C**TG**T**A**C**T** ; S_2: **TA**G**C**T**C** ; S_3: C**TGACT**TTC

⇒ La longueur du LCS est de **4**
Les 4-LCS sont : **TACT** , TGCT , et TGTC

Remarque: A noter que l'alphabet n'est pas limité à ces deux exemples, c'est un paramètre à faire varier durant les expérimentations.

Le problème de **la sous-séquence commune la plus longue** faisant l'objet de ce projet, il vous est demandé :

- Modélisation du problème (solution, fonction d'évaluation, etc)
- Génération des Datasets (varier l'alphabet, nombre de séquences dans S, taille des Séquences S_i).
- Résolution à l'aide d'une méthode exacte (DFS, BFS, ..au choix).
- Résolution à l'aide d'une métaheuristique : affectation aléatoire*.
- Expérimentation sur les différents datasets.
- Comparaison et discussion des résultats obtenus (critères de comparaisons: temps d'exécution, meilleure évaluation trouvées).

Rapport

Chaque binôme devra remettre un rapport détaillé du travail effectué avant la date butoire du **29/04/2024**. Le rapport de **maximum 20 pages** devra au moins contenir :

- Introduction,
- Définition et modélisation du problème (avec exemple),
- Fonctionnement et Algorithme des deux approches (avec exemples),
- Expérimentation sur la taille du problème pour chaque approche,
- Comparaison (Graphes/Tableaux, Analyse et discussion) des approches,
- Conclusion.

Implementation

- Le langage de programmation est au choix (Python, Java…).
- Les deux algorithmes doivent être développés from scratch (Interdits d'utiliser des algorithmes déjà implémentés ou librairies).

Evaluation

- (*) Une métaheuristique sera affectée à chaque binôme prochainement.
- Travail en binôme.
- Projet noté sur 20 Pts :
  - Rapport sur 10 Pts.
  - Demo Code sur 10 Pts.
- Rapport à remettre en version papier + par email.
  - n.a.houacine@gmail.com
  - belkadi.wh.usthb@gmail.com

Bon courage.